FLUKE networks

Improve Networked Application Performance Through SLAs

Table of Contents

*Fluke Networks would like to acknowledge **Dr. Cathy Fulton**, **Ph.D.** of **NetQoS, Inc.** for her substantial contributions to this document.*

Executive summary

Successful Service Level Management (SLM) requires a proactive approach, yet today most management teams primarily operate in reactive mode. "Improve Networked Application Performance Through SLAs" describes how to transform your team from firefighters to advanced planners with the appropriate tools.

SLM is receiving widespread attention as a method to align IT resources with business goals. SLM is a process for controlling the quality of a delivered service in order to consistently meet client requirements and continuously improve operational efficiency. It provides a means for IT to be measured on return on investment rather than total cost of ownership.

This paper discusses the considerations and decisions needed to implement a successful SLM strategy, detailing how to define Service Level Agreements (SLAs) using the appropriate Service Level Objectives (SLOs) to accomplish business goals. The paper reveals the common pitfalls of SLAs and how to avoid them; explores the details of selecting and measuring SLOs; and discusses how to operate proactively using existing tools.

Proper selection of the SLOs is critical to success. This paper describes how to select the metrics to measure, whether they are focused on the client, server or network. It explains the practical impact of choosing the statistics, averages or percentiles on the management strategy, and details how to select the thresholds that determine compliance. An evaluation of the different monitoring solutions is also performed, highlighting strengths, weaknesses and the potential business impact.

With SLM on the horizon, find out how to successfully implement the right SLAs for your enterprise.

Introduction

Traditional Service Level Management (SLM) is based solely on availability monitoring. The service (network, server or application) must be "up" 99.999% of the time. This metric is easy to understand and seems to offer real value to end users. However, it fails to satisfy key SLM objectives, client requirements and continuous improvement. It does not meet client requirements, for a service that is "up" may have such poor performance as to be rendered unusable. Nor does it facilitate continuous improvement in operational efficiency; rather, it places the management focus on events that rarely occur.

To maximize effectiveness, SLM must be solidly founded on performance, in addition to availability. It must be implemented with the end users' experience in mind, not only the infrastructure status. And, it must do so in a manner that creates, rather than consumes, time. Fortunately, SLM tools have finally evolved to the point where these requirements can be satisfied. Today's SLM tools should actively encourage an evolution from reactive to proactive management by providing functionality in four key areas: multi-tier reporting, early detection, rapid resolution, and opportunity discovery. In addition, this functionality should be pack-

aged in an easy-to-deploy, easy-to-manage, and easy-to-use architecture.

SLM requires careful definition of the Service Level Objectives (SLO) in order to be effective. There are three key performance variables that should be measured: end-user response time, server response time, and network delay. But how these three are measured, passively or actively, can dictate success or failure in achieving the desired results. The SLOs may be based on time averages, on percentages of time averages, or on transaction percentages. While many tools on the market allow for tracking a SLO based on time average, this method has a drawback that the average may not necessarily reflect what the majority of users are experiencing. SLO tracking based on transaction percentages is a technically superior method, and it accurately captures the user experience. However, few solutions exist to effectively implement this method across an enterprise.

Configuration of the thresholds used in the SLO should be based on user requirements. These requirements vary both by application and by network access method. Generally two thresholds should be specified. The first, or lower threshold, should reflect the point at which users become dissatisfied. The second, or upper threshold, should reflect when poor system performance causes significant business cost. The percentages, if the SLO supports them, should be adjusted over time to drive continuous operational improvement and to control delay variation.

Proactive management through SLM

Typically, a change in mindset is necessary to successfully accomplish the true objectives of SLM. Most IT teams currently operate in a reactive mode. Much of their time is dedicated to dealing with crisis management, desperately trying to contain and extinguish fires. By managing IT resources through SLM, IT departments can anticipate problems and rapidly resolve the issues, taking them from a reactive to a proactive team. This shift in behavior patterns will no doubt require departmental training, but the right tools can provide a critical jump start to help people evaluate network performance from a new perspective.

True SLM tools are much more than methods of monitoring and analysis. They ensure that the necessary resources are being applied in alignment with the needs of the business users. The first requirement of the tool is that it must free up time for strategic action. Some tools are so cumbersome to deploy, manage, and use that there is not a significant time savings for the IT team. The SLM tool selected must be easy to use and must deliver functionality to be truly effective.

The ease of use of a SLM tool represents the degree of effort required for it to be deployed, managed, and used. This is determined by the architecture of the SLM tool in conjunction with the particulars of the destined environment. A tool that is virtually unmanageable in a global enterprise may be quite acceptable for a smaller environment. A tool that is exorbitantly expensive for a mesh network may be reasonably priced for a hub-and-spoke environment. A tool that requires continuous coordination among different IT teams (e.g., those managing desktop support or wide-area network applications) may be a source of intense stress or an opportunity…but it is usually a source of stress and inefficiency.

A SLM tool must actively encourage the move from reactive to proactive management. It accomplishes this by providing functionality in four key areas: multi-tiered reporting, early detection, rapid resolution, and opportunity discovery. These areas are discussed later in this document.

Valuable variables

One of the first decisions in deploying a SLA involves selection of the variables. On what variables should the SLA be based? There is often a conflict between what the end user desires and what the IT team can deliver. The end user wants a metric that is directly meaningful – typically end-user response time. The IT team wants a metric they can manage (e.g., if they do not control the server farm, they do not want to be held accountable for server issues). A good compromise is to measure broadly, but blame selectively; to monitor the commonly understood variables, but restrict penalties according to responsibilities.

End-user performance

The end-user response time of an application should be monitored regardless of the existence of a SLA. This variable provides insight into the end-user mood, and facilitates meaningful communication between the IT team and the user. The most common method of expressing end-user experience is through measuring transaction times and their components.

The real decision when trying to measure the end-users' experience is determining which and how transactions are measured. Should every different transaction be monitored or only a select few? The former requires aggregation for scalability, resulting in some loss of visibility. The latter requires diligence in ensuring that the few selected transactions are current, representative, and important. A combination of the two methods often yields the most satisfying results. That is, the two methods need not be mutually exclusive.

Should real users be passively monitored, or should synthetic agents be activated? The former is absolutely essential to achieving the goals of SLM. The latter can provide a deterministic baseline that is useful for troubleshooting. The best approach is to

combine real-user, passive monitoring with a handful or fewer synthetic agents; in this manner, the benefits of both approaches can be effectively realized.

Server performance

The server response time should also be monitored regardless of the SLA. It is very useful to be able to quickly determine whether the servers are the problem if the end-user response time deteriorates. This metric can also be used to track the quality of service delivered by the data center. The server response time is also essential for optimization and planning activities.

There are some serious issues associated with how the server response time is measured. If synthetic agents are used to repeatedly run the same transactions, the results may be cached either by the client or the server. This caching effectively invalidates the results since it is not representative of the real user experience. If the server is caching the information, it can not be selectively disabled. If the transactions are randomized, then the main benefit of synthetic agents – their determinism – is lost. This selective caching effect can render synthetic agents inaccurate for measuring server response time. Passively monitoring server performance for all transactions and all system users eliminates these problems and can provide a useful baseline for future performance.

Network performance

Network delay is another metric that should be monitored regardless of the existence of a SLA. In the same respect as server performance, network performance is very useful in quickly determining whether the network is the problem if the end-user response time deteriorates. Network performance metrics – such as round trip time – can be used to measure the level of service received from a network provider. Continuous monitoring of network delay is also essential for optimization and planning activities.

There are several common methods for measuring network delay. Active methods include scheduling ICMP pings or TCP session connects. Passive methods include measuring TCP session connects or more general application packets. Of each of these methods, network delay measurements based on observing general application packets provide the most accurate representation of performance. It is important to understand the network delay components in order to appreciate the merits and limitations of each approach. The network delay consists of five components: serialization, queuing, propagation, processing, and protocol delay as described below.

Serialization or transmission delay is the time required to put all the bits in the packet on the transfer medium. It is dependent on both the packet size and the link access rate. A 64-byte packet will have a round-trip serialization delay of 18.3 ms on 56 Kbps circuits, 4.0 ms on 256 Kbps circuits, and 0.7 ms on 1.5 Mbps circuits. A 1500-byte packet will have round-trip serialization delays of 428.6 ms, 93.8 ms, and 16 ms respectively. TCP session connects involve 64-byte packets. As a result, measurements based on TCP session packets will generally underestimate the network delay experienced by the rest of the application. ICMP pings can be configured to assume any size, but the packet size is always the same in both directions. Most applications do not have this symmetry, which makes it difficult for ICMP to accurately capture the serialization delay experienced by the application. Note also that the default ICMP packet size is 64 bytes.

Queuing delay is the time the packet waits in a buffer for its turn to be transmitted. It depends on the serialization delay for the packets served ahead, the dimension of the buffers, the amount of congestion, and the configuration of the router or switch scheduling policies. Congestion can change dramatically in microseconds, but a TCP session may be open for seconds or hours or even days. Thus the queuing delay experienced by the TCP session connects can be significantly different from that of the main application. The same is true for any scheduled probe like ICMP; the queuing delay even 60 seconds earlier may bear little resemblance to that experienced by the application. Additionally, the router or switch may place ICMP packets in a special queue for preferential (either better or worse) handling. During periods of congestion, ICMP packets may be preferentially dropped while the application packets wait – thus ICMP never measures the longer delays. ICMP packets may be preferentially moved to the head of the queue, thus experiencing shorter delays; they may be selectively moved to the rear of the queue, thus experiencing longer delays (unless dropped).

Propagation or distance delay is the time it takes the packet to travel along the physical path. It is dependent only on distance and the type of medium. If the TCP session connects and ICMP packets travel the same physical path as the main application packets, then the propagation delays will be identical. However, it is not guaranteed that the same paths will be traversed, which if true, would render the ICMP measurement irrelevant.

Processing delay is the time it takes the router or switch to prepare the packet for delivery. It is dependent on a wide variety of factors, but it is normally insignificant. Note that TCP session connects may require more processing than the remaining packets in the flow, and ICMP packets may require less processing.

Protocol delay is the time the packet waits due to the underlying protocols. For example, in a shared medium, the packet must wait for its node to acquire access. The effect of this delay varies greatly with protocol.

In summary, network delay measurements based on ICMP pings only reveal the delay experienced by ICMP pings at that snapshot in time. Network delay measurements based on TCP session connects only reveal the delay experienced by 64-byte packets at the time the session was established (seconds, hours, or even days ago). Of each of the methods, passively observing general application packets is the most effective means for measuring network delay, as it reflects what users are actually seeing.

Service availability

Service availability should be explicitly monitored as part of an SLM strategy. Traditional approaches to fault management have called for the tracking of network and server device availability. This can be augmented with active agents or probes that can periodically test select transactions. If the probes are scheduled to run every 15 minutes, a sustained outage can be detected on average 7.5 minutes after it begins. However, intermittent brief outages would go undetected and not be tracked against the SLO. More frequent polling could detect shorter outages, but this would come at the expense of placing additional load on the system.

Slippery statistics

The next important decision when implementing SLM, whether realized or not, involves statistics. Should the SLA be based on time averages or transaction percentages? A SLA based on time averages would require, for example, that the average end-user response time be less than 3 seconds. A SLA based on percentiles would require, for example, that 95% of the transactions have response times less than 3 seconds.

The advantage to choosing a SLA based on time averages is that nearly every SLM vendor supports averages, providing great freedom in tool selection. Unfortunately, time averages do not provide a representation of what the users are experiencing. For example, suppose nine users each observe a 0.5 second response time, while the tenth user receives a 90.0 second response time. The reported average response time is 9.5 seconds – which differs by an order of magnitude from what any user actually experienced. Because of this sensitivity to skew, it can be very difficult to manage to an average. If the tenth user received a response time of 180.0 seconds (rather than 90.0 seconds) while the other users remained constant at 0.5 seconds, the average would nearly double – even though only one user experienced performance degradation.

Some vendors report a trimmed average to reduce this sensitivity to skew; they discard any measurement that is above a pre-set threshold. In the previous example, a pre-set threshold of 2 seconds would result in a trimmed average of 0.5 seconds. The danger with this approach is that it can mask very real performance problems. If a network issue develops such that the response times experience by seven users increased from 0.5 to 2.5 seconds, the reported trimmed average would remain 0.5 seconds – even though 80% of the users suffered performance degradation. Given the heterogeneous nature of most environments, selection of an appropriate trimming threshold is nearly impossible. Indeed, there have been cases where the worst-performing sites were reported as the best performers due to the trimming.

A SLA based on transaction percentages is impervious to skew and relates directly to the user experience. If 95% of the transactions have a response time less than 3 seconds, the values of the remaining 5% are not significant. A SLA based on trimmed averages ignores all response times exceeding a pre-set threshold; if all response times exceed the threshold, there is no measurement. A SLA based on transaction percentages ignores a pre-set percentage (in the example, 5%) of response times.

A SLA based on transaction percentages is preferable to that based on time averages; however, the choice in SLM vendor is more limited. It is technically more challenging to monitor and report percentages compared to averages, so fewer vendors support this option. Some vendors choose a hybrid approach of reporting percentages of averages (rather than percentages of transactions). A SLA based on this hybrid approach would require, for example, that 95% of the 5-minute averages during the month must be less than 5 seconds.

In summary, a SLA may be based on time averages, on percentages of time averages, or on transaction percentages. A SLA based on time averages has issues with skew; the results may not be representative of the users' experience. SLAs based on transaction percentages are technically superior, but have not been as widely implemented.

Defining details

Another important decision involves identifying the actual objectives. How many objectives will there be for each variable? What timeframe should be used in determining compliance? What thresholds and percentages are appropriate? These defining details should be solidly based on user expectations to accurately measure the end-users' experience.

There are two natural thresholds of interest: that of insignificance and that of pain. Delays that are less than the "threshold of insignificance" are not noticed by the user. This does not necessarily imply that the delays are negligible – only that the delays fall well within the users' expectations, that they do not generate any annoyance. Delays that are greater than the "threshold of pain" result in user abandonment. Such delays are expensive in terms of lost business opportunity or employee productivity. Delays that fall

between the thresholds typically result in minor griping about application sluggishness.

These two natural thresholds are typically not known, but can be discovered through experimentation (with cooperative or unwitting users, depending on politics). Some generic values frequently cited for web page downloads are 3 seconds and 8 seconds. However, thresholds generally depend on the network access method and the application itself. For example, users accessing an entertainment portal over satellite will have a greater tolerance for delay than those accessing helpdesk requests over a DS3 terrestrial link. A separate SLA should be defined for each application and access grouping.

The thresholds should be established based on user requirements. The percentages, if the SLA supports such, should be adjusted to drive continuous operational improvement. Users tend to be sensitive to variations in delay, not just their absolute values. Increasing the percentages effectively controls the delay variation. As an example, suppose the SLA initially states that 95% of transaction response times must be less than 3 seconds and 98% must be less than 8 seconds. The goal should be to increase these percentages to, say, 96% and 99%, respectively, over some time interval. Decreasing the 3 second threshold would have little business impact since it is already at an acceptable value.

It may be desirable to specifically exclude maintenance windows, or even specific users, from the SLA. This should be determined in the definition phase – rather than after the SLA is not achieved. Note, however, that few vendors currently support such features. If the selected vendor does not support desired exclusion windows, then the defined percentages should be adjusted downwards in a compensatory fashion.

In summary, the thresholds used in the SLA should be based on user requirements. These requirements vary both by application and by network access method. Generally, two thresholds should be specified. Delays below the lower threshold have no user impact; delays above the upper threshold have significant business cost. The percentages, if the SLA supports such, should be adjusted over time to drive continuous operational improvement and to control delay variation.

Selecting the right SLM solution

As mentioned earlier, a SLM initiative must actively encourage the shift from reactive to proactive management. The solution implemented to automate SLM must provide functionality in four key areas: multi-tier reporting, early detection, rapid resolution, and opportunity discovery. These areas are discussed in detail in the following sections.

Multi-tier reporting

Many vendors claim that their tool enables SLM, leaving the interpretation and implementation up to the user. Certainly a packet trace enables SLM, but it is not always practical within established time constraints. Nor is a tool particularly useful if it only provides high-level "management," but lacks the detail necessary for taking appropriate action. A SLM tool should

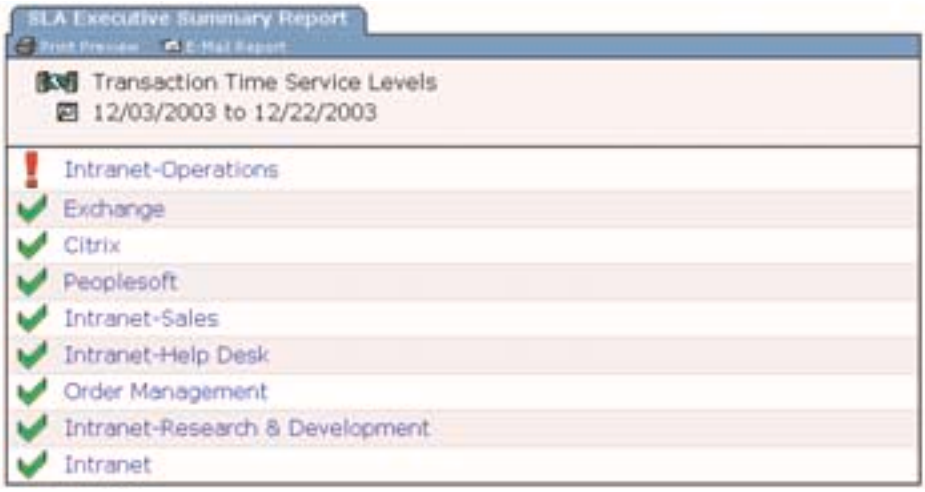

Figure 1. Top-tier SLA report.

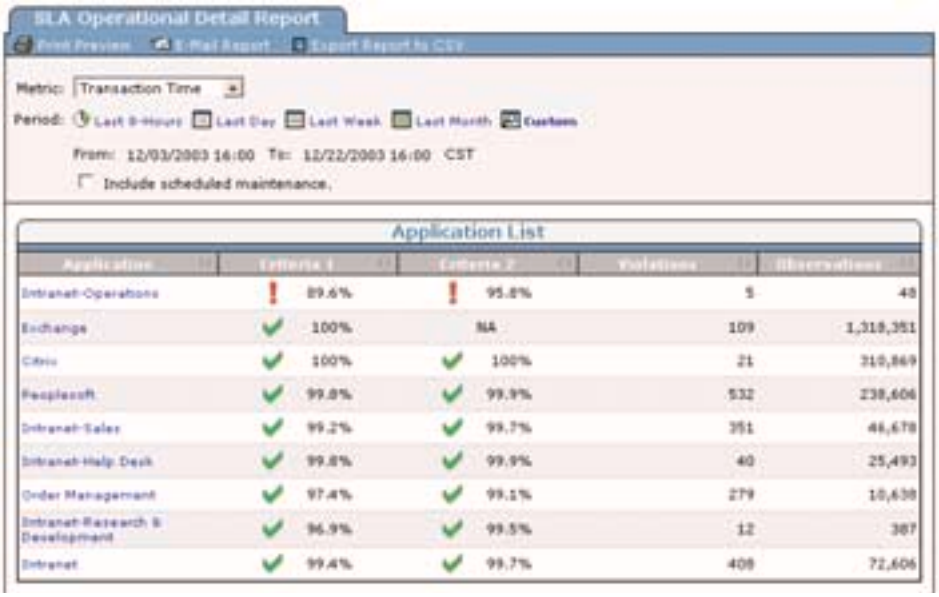

Figure 2. Top-tier SLA report - PeopleSoft.

provide easy navigation from high-level status to technical level detail, as painlessly as possible. In short, it should provide multi-tier reporting. High–level summaries are primarily useful for communication with the non-technical audience, as well as a navigation aid to quickly reach the relevant technical detail.

Top-tier SLA reports

The top-tier SLA report in Figure 1 provides an at-a-glance compliance status of the various SLAs for business users. Should more detail be required, any application name can be clicked to drill-down into a more detailed compliance report specific to that application.

In Figure 2, the compliance metrics of the Peoplesoft application are shown. This SLA requires that 95% of all Peoplesoft transactions have response times of less than 2 seconds (Criteria 1) and 99% have response times less than 4 seconds (Criteria 2). The Peoplesoft service is in compliance with this SLA because 99% of transactions are less than 4 seconds, and only 99.8% are less than 2 seconds.

Figure 3 represents a view of compliance more appropriate for IT management or technical users. This view provides observation and violation counts, as well as more reporting options to change the information contained in the report. Top-tier reports are very useful for spotting issues and violations, but they do not provide enough information to suggest any viable course of corrective action.

Mid-tier SLA reports

Mid-tier SLA reports provide different temporal, spatial, or logical summary views of the SLA compliance. For example, Figure 4 shows SLA compliance as a function of time, allowing periodicities to be readily spotted or problem intervals to be more deeply investigated.

Alternative views should also be provided to determine if an individual server or specific group of users is contributing an undue amount of violations. For example, if SLA violations are caused by a few client sites, this will be evident in the view of client regions illustrated in Figure 5. These views are essential in helping the IT group understand how to bring the application into compliance.

Lower-tier reports

Lower-tier reports are essential for rapid resolution of performance problems that arise, in addition to aiding in the effective allocation of IT resources. They provide the necessary detail required for understanding the scope and cause of the problem, enabling IT staff to take action on the reported issues. These lower-tier reports include the results of automated investigations (Figure 6), as well as detailed performance metrics and statistics (Figure 7).

Intelligent baseline reports

In addition to tracking performance against a static SLA threshold, it is important to understand how current performance

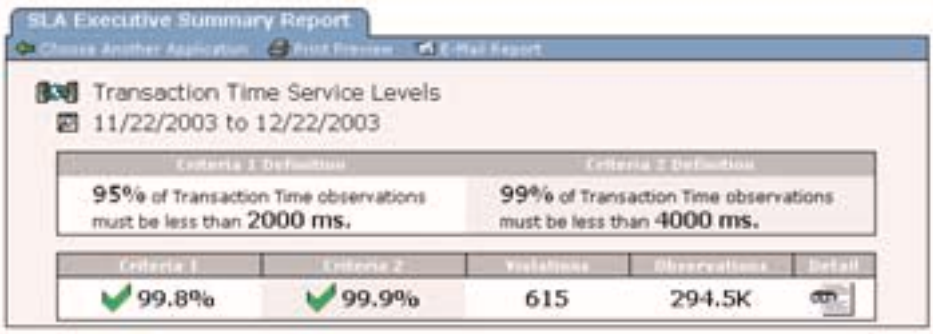

Figure 3. Top-tier SLA Report - all applications.

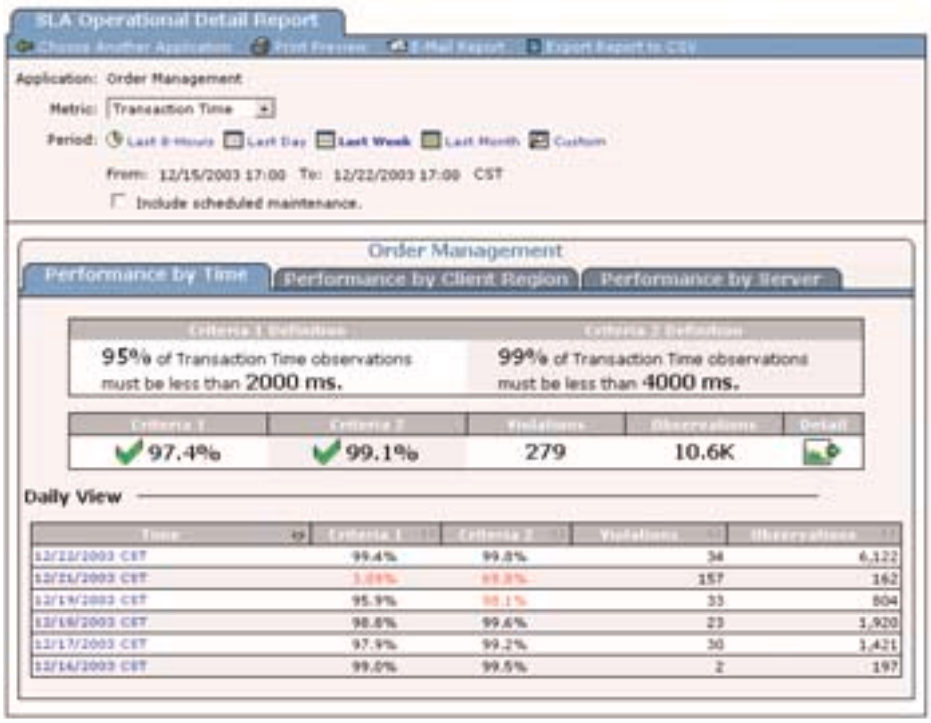

Figure 4. Mid-tier SLA report - by day, order management system.

compares to past performance. Users' expectations are set by their previous use of an application – you may be well within the constraints of your SLA, but still encountering upset users, based on slower response times than they are used to. This type of report can be generated, provided an intelligent baseline has been calculated for application performance. The baseline should take into account recent and historical system performance. Figure 8 shows a top-tier view of how each application is performing against its historical baseline. Figure 9 illustrates a mid-tier view of how performance for a Citrix application over the past eight hours compares to past performance.

Early detection

Everyone is familiar with the most common methods for discovering issues and crises in an enterprise: the phone rings or an urgent e-mail is received. Most IT teams do not have time to dedicate each time an upset individual runs into their office. Unless issues are detected early, the team will spend much of its time fighting fires and doing little to address the long-term needs of business users.

A SLM tool must have a method of automatically discovering smoldering embers before they escalate into five-alarm fires. This automated discovery mechanism, coupled with prioritized reporting, is absolutely critical for proactive operations. While older tools rely on pre-configured static thresholds to detect issues, a new generation of tools use self-learning algorithms. The tools learn "typical" behavior for applications, servers, and client regions while capturing the normal daily, weekly, and monthly periodicities. They understand that the last Friday in a month may naturally be slower than other time periods; they will not generate an alert unless the behavior is poor compared to the learned norm for this time period.

Intelligent baselines automate the discovery of developing issues, alerting the IT team to potential problems before users are significantly impacted. This early discovery reduces mean time to repair (MTTR), increases productivity, and enhances the team's reputation. The newer tools can search across the enterprise, looking for anomalies, inefficiencies, and other areas for

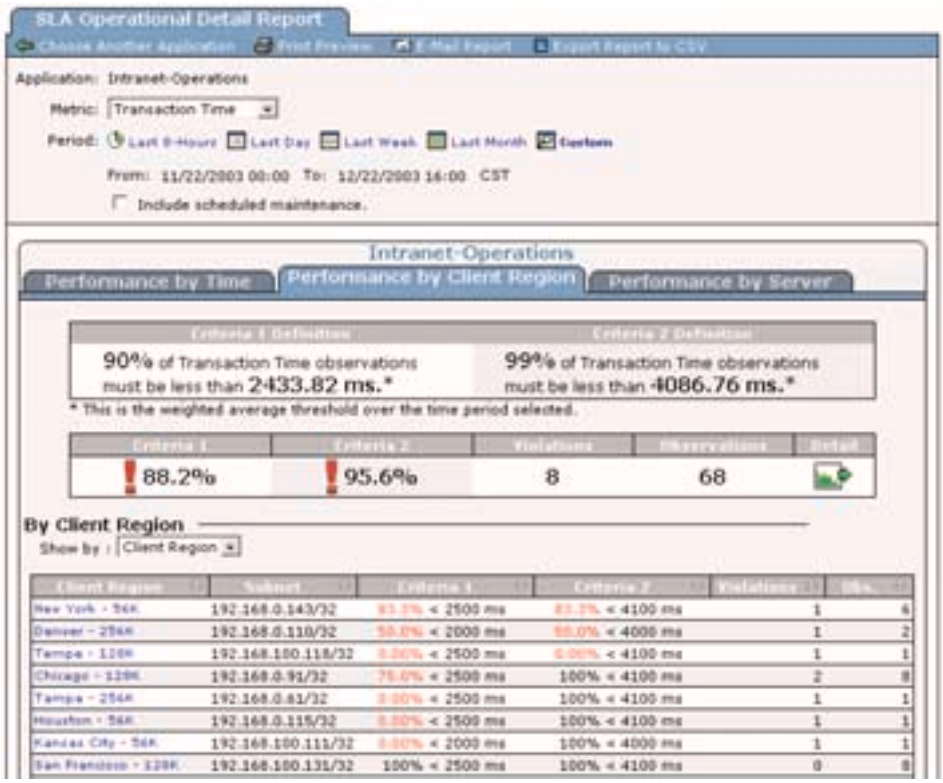

Figure 5. Mid-tier SLA report - by network subnet/user group.

Figure 6. Lower-tier investigation report.

improvement. They provide 24x7 monitoring and analysis of the performance data that is arriving.

Figure 10 provides an alternate top-level view of performance – detailing the most critical performance incidents detected over the past two weeks.

It is essential to recognize availability as well as performance issues. Active monitors are typically used for such a function, but they do have several drawbacks. In the standard implementation, active monitors test availability (and performance) periodically. They are scheduled to run every 5 or 15 or 30 minutes. If the agents are scheduled to run every 15 minutes, on average they will detect an outage 7.5 minutes (but possibly 15 minutes) after it occurs. The shorter the scheduled interval, the more quickly the agents can detect an issue – but the greater stress they place on the network and servers. Because of this stressor, active monitors can only test select transactions from select locations. It is all too common for agents to induce an event they were intended to detect.

A better approach is to combine passive monitoring with triggered active investigations. Only if an unusual absence of traffic is detected will the network or server be actively probed – and at that time, the stress is minimal if there is not a real outage. Using this method, an outage may be detected quickly without placing additional load on the network or servers.

Regardless of the actual implementation, early detection of performance and availability issues is a fundamental component of SLM.

Rapid resolution

The SLM tool chosen must not only detect developing issues, it must also assist in their rapid resolution. Multi-tier reporting certainly facilitates this, particularly when it is integrated with a click-and-browse navigational interface. Form-based custom reports are wonderful for flexibility, but they

Figure 7. Lower-tier response time components.

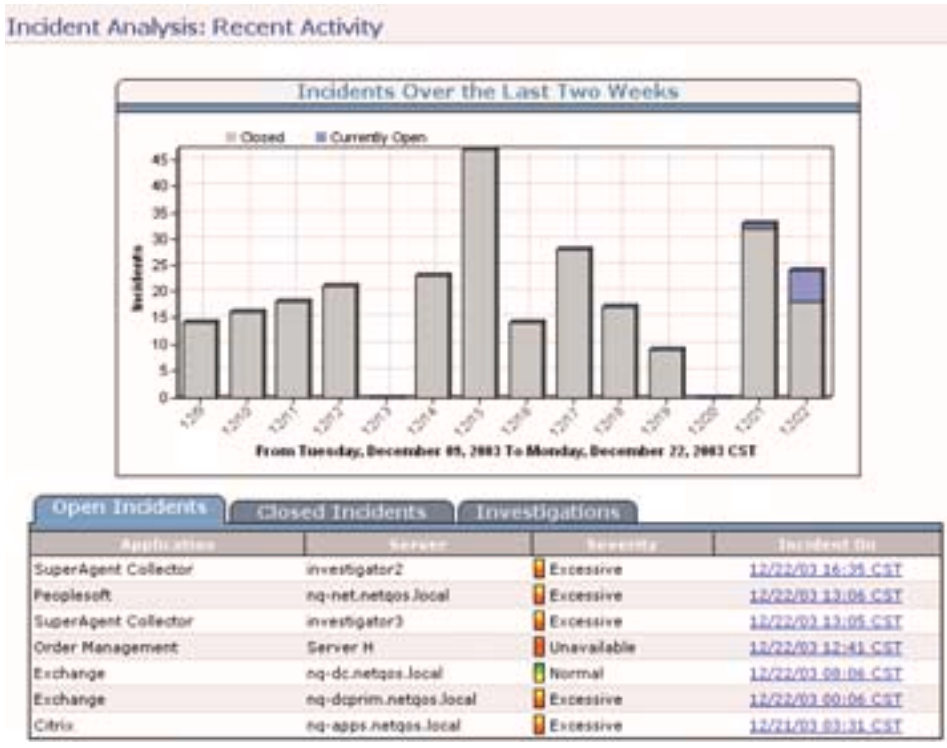

Figure 8. Top-tier intelligent baseline report.

provide a painful and tedious interface. They are much better as supporting cast rather than the lead.

Automated investigations can be significant time-savers, so long as they require little manual configuration. When a developing server problem is detected, additional information such as CPU utilization, memory usage, and top processes should be gathered – at the time the issue is occurring. When a developing network problem is identified, trace routes should be launched or additional MIB statistics collected. Such triggered investigations can save on much of the diagnostic legwork.

Continuous improvement

One of the main objectives of SLM is continuous improvement. Certainly the early detection and rapid resolution of issues improves operational efficiency. However, these activities are still reactive in nature. The service must already be unacceptable (via the SLA threshold) or it has begun deteriorating (detected by intelligent baselines) to trigger action. If the service is in a steady but inefficient state, it will not be noticed. A SLM tool should provide a mechanism to quickly discover these inefficiencies, and identify opportunities for improvement.

An example of such a feature appears in the reports shown in Figures 11-15. These performance maps provide high-level views that are extremely useful for improving performance. The maps allow you to choose from a number of options, including the application(s), client region(s), server(s), metric of interest, sort order, and time period.

The following three subsections provide examples of how these performance maps can be utilized effectively. “Application inefficiencies and opportunities” shows how performance maps can reveal the interactions in a multi-tiered application. “Network inefficiencies and opportunities” reveals how performance maps provide enterprise latency

Figure 9. Mid-tier intelligent baseline report.

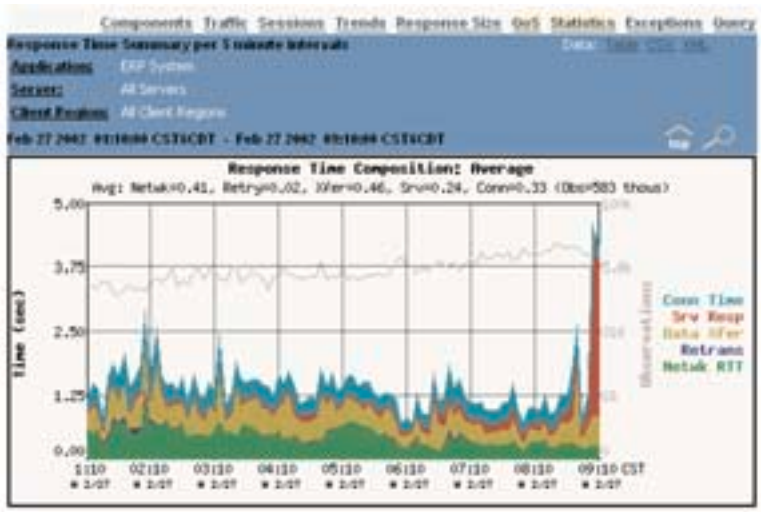

Figure 10. Top-tier incident report.

maps, traffic volume matrices for capacity planning, and prioritization of problem sites. "Server inefficiencies and opportunities" describes how performance maps can identify problem servers and ineffective load balancing.

Application inefficiencies and opportunities

Figure 11 depicts the "transaction time by application" performance map for a globally deployed multi-tier Enterprise Resource Planning (ERP) application. SuperAgent monitors each tier of this application: web GUI (ERP system), user authentication (LDAP directory), document exchange (NetBios/TCP), and back-end database (Oracle 9i DB). The GUI application naturally has the largest average transaction time (1.51 seconds) while the backend database has the smallest average transaction time (0.04 seconds); user authentication imposes a delay of 0.53 seconds. This performance map provides a quick snapshot of how each application tier is behaving and if one is affecting another. If both the GUI and database times were high, then it is likely that one was affecting the other. In this case, the user would click on an application name to drill-down into a lower-tier detailed report to see the correlation between the two, and to identify the source of the issue.

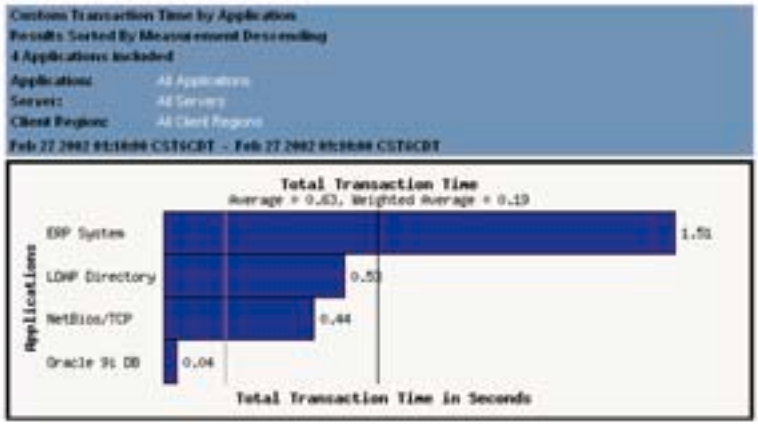

Figure 11. Transaction times for multi-tiered application.

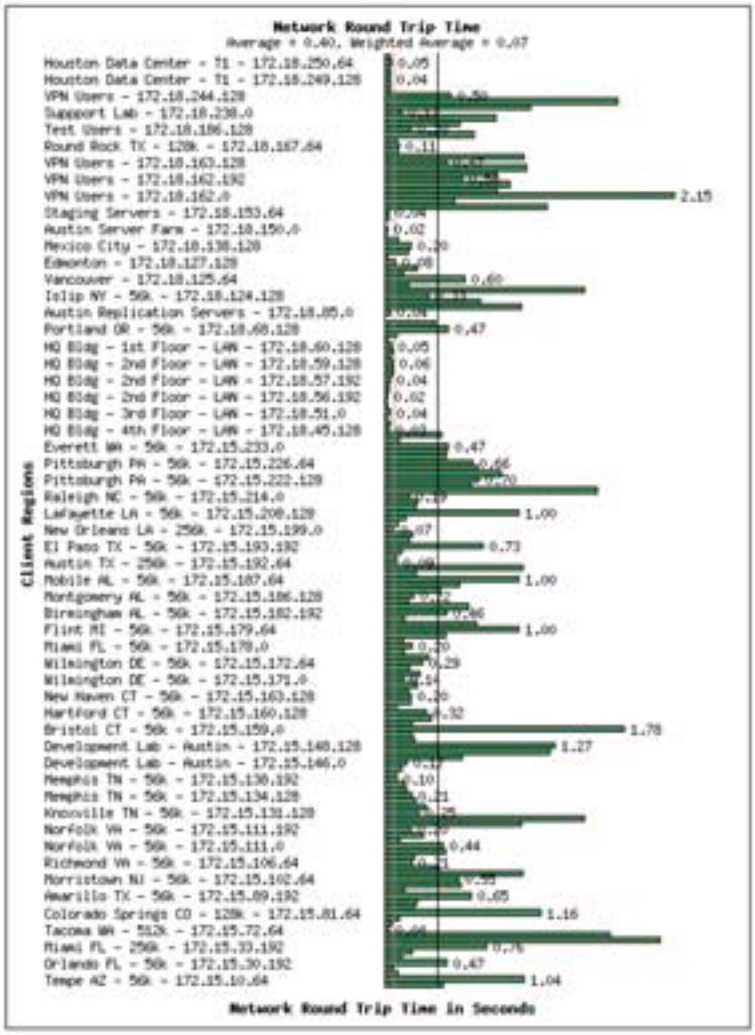

Figure 12. Network latency map: finding hot spots.

Network inefficiencies and opportunities

Performance maps can be used to create latency and loss maps of the network. Figure 12 shows the "network round trip time by client region" graph, giving an at-a-glance view of network performance across the entire enterprise. All sites are included and sorted by description to provide visual identification of network hot spots. For example, VPN users have experienced poor perform-

ance when compared to others, while all users at the corporate headquarters enjoy fast performance.

The "percent byte loss by client region" performance map in Figure 13 reveals the top 15 client regions with the worst percent byte loss (sorting is by metric rather than by description or by address). High loss rates may be caused by errors or congestion; in either case, they represent significant inefficiencies and opportunities for improvement. The productivity of users in Pittsburgh and El Paso are severely limited because of the network conditions.

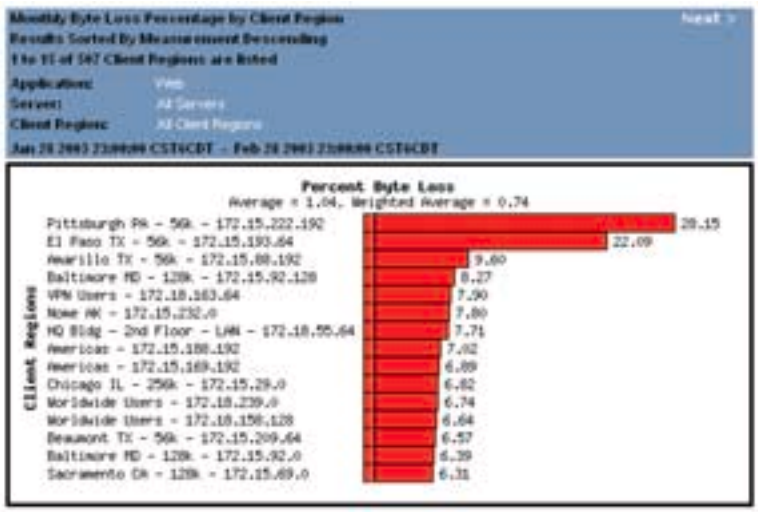

Figure 13. Network loss map: top worst sites.

Server inefficiencies and opportunities

Performance maps can be used to identify problem servers by comparing performance across the members of a server farm. The "refused sessions by server" performance map in Figure 14 shows that ERP Server 1 is overloaded or malfunctioning. The "server response time by server" performance map in Figure 15 illustrates that the web servers are providing inconsistent service levels, with the fastest providing response times that are seven times faster than the slowest. These may be older systems requiring upgrades or a load balancing issue. Performance maps can evaluate the effectiveness of load balancers by comparing the number of active sessions, the traffic volumes, and the response times. Different implementations use different balancing metrics. Performance maps can also assist with internal server farm optimization by providing traffic volume matrices between the systems.

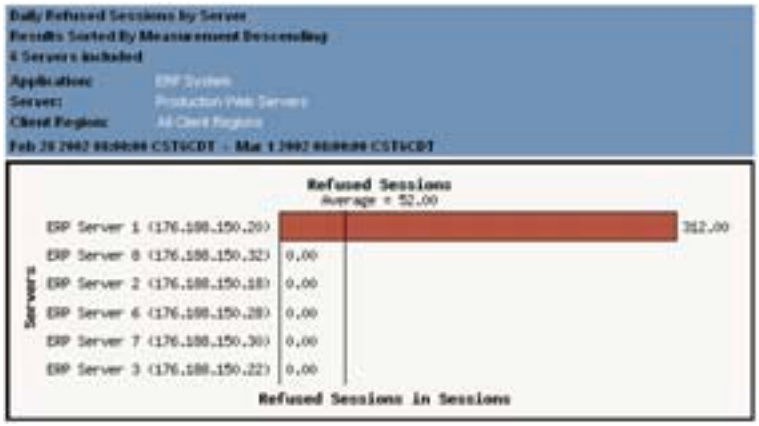

Figure 14. Server inefficiencies: top refused sessions.

Conclusion

Service Level Management is a process for controlling the quality of a delivered service in order to consistently meet client requirements and continuously improve operational efficiency. Since the clients of IT are the end users and the job of the IT department is to facilitate these users conducting business, SLM can be looked at as a method of ensuring that IT is aligned with business success.

When adopting a SLM program, there are two requirements for success: the technical objectives must be carefully defined and the team must learn to operate strategically. When defining the technical objectives, the services to monitor, metrics to measure, method of measurement, and tools available to deploy SLAs must be taken into account. The selected SLM tool should encourage proactive management by providing functionality in four key areas: multi-tier reporting, early detection, rapid resolution, and opportunity discovery. Moving team operations from firefighting mode to strategic planning requires the successful implementation of the technical objectives and the integration of SLM into daily practices.

SLM enables IT professionals to adopt cycles of continuous improvement in the services they provide to the business. Analysis of past performance and compliance allows IT staff to identify areas of improvement that will provide the highest impact to service levels. The resulting alignment of IT resources and initiatives with business performance is a high-value benefit for any enterprise.

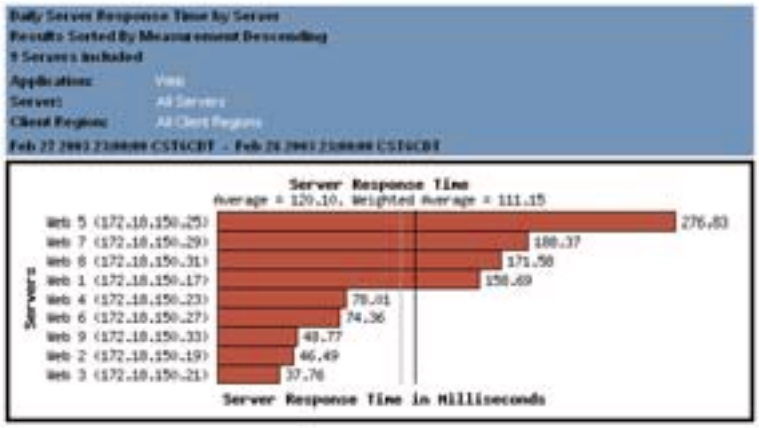

Figure 15. Server inefficiencies: load balancing.

NETWORKSUPERVISION

Fluke Networks
P.O. Box 777, Everett, WA USA 98206-0777

Fluke Networks operates in more than 50 countries worldwide. To find your local office contact details, go to **www.flukenetworks.com/contact**.

Printed in U.S.A. 2/2004 2132182 A-ENG-N Rev A